Applicant Initiated Interview Request Form

Application No.: 09/ 542,525 First Named Applicant: __Kuch__
Examiner: __Qamrun Nahar__   Art Unit: __2124__   Status of Application: First OA

Tentative Participants:
(1) _Gregory L. Maurer_   (2) _Justin D. Wagner_

(3)_____   (4)_____

Proposed Date of Interview: 07-13-2004   Proposed Time: 2:00 EDT (AM/PM)

Type of Interview Requested:
(1) [X] Telephonic   (2) [ ] Personal   (3) [ ] Video Conference

Exhibit To Be Shown or Demonstrated: [ ] YES   [X] NO
If yes, provide brief description:_____

Issues To Be Discussed

Issues (Rej., Obj., etc)	Claims/ Fig. #s	Prior Art	Discussed	Agreed	Not Agreed
(1) _Rej._	_Cl. 1_	_Chilimbri_	[ ]	[ ]	[ ]
(2)_____	_____	_____	[ ]	[ ]	[ ]
(3)_____	_____	_____	[ ]	[ ]	[ ]
(4)_____	_____	_____	[ ]	[ ]	[ ]

[ ] Continuation Sheet Attached

Brief Description of Arguments to be Presented:
Chilimbri's description of placing objects fails to teach or suggest grouping data members of an object class into a plurality of separate groups and assigning data members within first and second units of memory of a virtual memory system, the second unit being separately loadable from the first unit. SEE ATTACHED SHEETS

An interview was conducted on the above-identified application on _____.

NOTE:
This form should be completed by applicant and submitted to the examiner in advance of the interview (see MPEP § 713.01).
This application will not be delayed from issue because of applicant's failure to submit a written record of this interview. Therefore, applicant is advised to file a statement of the substance of this interview (37 CFR 1.133(b)) as soon as possible.

(signature) G. L. Maurer

_____   _____
(Applicant/Applicant's Representative Signature)   (Examiner/SPE Signature)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Kuch et al.

Application No. 09/542,525

Filed: April 4, 2000

Confirmation No. 5122

For: PROFILE-DRIVEN DATA LAYOUT
OPTIMIZATION

Examiner: Qamrun Nahar

Art Unit: 2124

Attorney Reference No. 3382-52327

PROPOSED REMARKS

Claim 1

Claim 1 is directed to a method of arranging a plurality of data members of an object class in a virtual memory system having separately loadable units, and recites in part:

> responsive to said consulting metadata indicating how the *data members of the object class are to be grouped into a plurality of separate groups* comprising a first group and a second group, assigning memory locations for *data members of the first group of the object class within a first unit* of memory in the virtual memory system; and
> responsive to said consulting metadata indicating how the data members of the object class are to be grouped into a plurality of separate groups comprising a first group and a second group, assigning memory locations for data members of the *second group of the object class within a second unit of memory* in the virtual memory system separately loadable into primary memory from the first unit. (emphasis added)

For example, the application describes at page 20, line 3, et seq.:

> An overview of an exemplary data layout optimization method is shown in Figure 5. At 502, profile data for the object is collected. Examples of profile data are shown in the illustrated embodiments below. At 504, the *data members of an object* are grouped based on the profile data. Techniques for achieving such grouping are shown below. Then, at 506, at runtime, data members from the same groups are arranged at neighboring locations in the memory system; members from different groups are placed at locations separately loadable from each other (e.g., members from Group A are placed at locations in one page of memory, and members from Group B are placed at locations in another page of memory). (emphasis added)

Thus, different groups of data members for the same object are placed at separately loadable memory locations. The Action rejects the claim based on Chilimbri. Applicants respectfully disagree.

Chilimbri's description of placing objects fails to teach or suggest grouping data members of an object class into a plurality of separate groups and assigning data members within first and second units of memory of a virtual memory system, the second unit being separately loadable from the first unit. In its rejection of claim 1, the Action relies on various passages in Chilimbri; however, these passages describe a scenario involving placing *objects*, not data members of the same object. Claim 1 recites "data members of the object class are to be grouped into a plurality of separate groups," "assigning memory locations for data members of the first group of the object class within a first unit," and "assigning memory locations for data members of the second group of the object class within a second unit of memory in the virtual memory system separately loadable into primary memory from the first unit."

For example, the Action relies upon Chilimbri at page 41, col. 1, par. 3, lines 1-6 to col. 2, par. 1, lines 1-5. During its discussion of layout of objects, Chilimbri states (as noted in the Action):

> As described in Section 3, generational garbage collection copies live objects to *TO space.* Our goal is to use data profiling information to produce a cache-conscious layout of objects in *TO space* that places *objects with high temporal affinity next to each other,* so that they are likely to be in the same cache block. The data profiling information captures the temporal ordering of base object addresses, which our system uses to construct object affinity graphs. An object affinity graph is a weighted undirected graph in which nodes represent objects and edges encode temporal affinity between objects. (emphasis added)

Thus, Chilimbri does describe "places objects . . . next to each other"; however, one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "data members of the object class are to be grouped into a plurality of separate groups," "assigning memory locations for data members of the first group of the object class within a first unit," and "assigning memory locations for data members of the second group of the object class within a second unit of memory in the virtual memory system separately loadable into primary memory from the first unit" from the mere mention of placing objects next to each other.